

## CLAIMS

What is claimed is:

1. A method for reducing processor cycles required to send data over a communication link in packets having a packet size, the method comprising:  
sending a write call comprising a destination and a quantity of data greater than said packet size to a driver through a socket;  
performing a zero-copy write of said quantity of data to said driver; and  
generating a plurality of packets less than or equal to said packet size.
2. A method according to claim 1, further comprising  
grouping data from a plurality of streams into said write call.
3. A method according to claim 2, wherein said grouped data comprises all data from said plurality of streams to be sent in a time interval.
4. A method according to claim 3, wherein the time interval is selected based on a bit rate of at least one stream.
5. A method according to claim 1, wherein said zero-copy write comprises writing data to a translation buffer.
6. A method according to claim 5, wherein the translation buffer is setup with a translation between virtual memory and physical memory.
7. A method according to claim 1, further comprising generating an interrupt only after a last packet of said plurality of packets is transmitted to said communication link.

8. A method according to claim 1, wherein said packet size is a maximum packet size allowable by the communication link.

9. A method according to claim 1, wherein the communication link comprises a network.

10. A method according to claim 1, wherein said write call further comprises a plurality of destinations.

11. A method according to claim 1, wherein said quantity of data comprises at least a portion of a multimedia data file.

12. A method according to claim 11 wherein the multimedia data file requires real-time delivery.

13. A method according to claim 11, wherein the multimedia data file is a video file, an audio file, or a game file.

14. A method according to claim 1, wherein said quantity of data comprises at least a portion of a file having a format chosen from the group of formats consisting of MPEG-1, MPEG-2, MPEG-4, H.264, MP3, QuickTime, AVI, Audio/Video, real-time data in RTP format, and combinations thereof.

15. A method according to claim 1, further comprising: communicating at least one of said packets to a network interface card.

16. A method according to claim 1, further comprising: storing said quantity of data in memory registered with said driver.

17. A method according to claim 1, wherein said write call comprises a write vector.

18. A method according to claim 1, further comprising generating a single header comprising header information for a plurality of protocol layers and sending the single header to a queue for a NIC.

19. A computer program product for sending data over a communications link in packets having a packet size, the computer program product comprising:

a computer-readable medium comprising a program module, the program module including instructions for:

receiving a write call comprising a destination and a quantity of data greater than said packet size through a socket;

performing a zero-copy write of said quantity of data; and

generating a plurality of packets less than or equal to said packet size.

20. A computer program product according to claim 19, further comprising instructions for:

grouping data from a plurality of streams into said write call.

21. A computer program product according to claim 20, wherein said grouped data comprises all data from said plurality of streams to be sent in a time interval.

22. A computer program product according to claim 21, wherein the time interval is selected based on a bit rate of at least one stream.

23. A computer program product according to claim 19, wherein said zero-copy write comprises writing data to a translation buffer.

24. A computer program product according to claim 23, wherein the translation buffer is setup with a translation between virtual memory and physical memory.

25. A computer program product according to claim 19, further comprising generating an interrupt only after a last packet of said plurality of packets is transmitted to said communication link.

26. A computer program product according to claim 19, wherein said packet size is a maximum packet size allowable by the communication link.

27. A computer program product according to claim 19, wherein said write call further comprises a plurality of destinations.

28. A computer program product according to claim 19, wherein said quantity of data comprises at least a portion of a multimedia data file.

29. A computer program product according to claim 28 wherein the multimedia data file requires real-time delivery.

30. A computer program product according to claim 28, wherein the multimedia data file is a video file, an audio file, or a game file.

31. A computer program product according to claim 19, wherein said quantity of data comprises at least a portion of a file having a format chosen from the

group of formats consisting of MPEG-1, MPEG-2, MPEG-4, H.264, MP3, QuickTime, AVI, Audio/Video, real-time data in RTP format, and combinations thereof.

32. A computer program product according to claim 19, further comprising: communicating at least one of said packets to a network interface card.

33. A computer program product according to claim 19, further comprising: storing said quantity of data in memory.

34. A computer program product according to claim 19, wherein said write call comprises a write vector.

35. A method for reducing buffering requirements on a network switch; the method comprising:

receiving a write call from an application through a socket, the write call comprising a plurality of destinations, including a first destination and a second destination, and a first quantity of data destined for the first destination, and a second quantity of data destined for the second destination;

packetizing said first quantity of data into a plurality of packets less than or equal to a packet size, each packet destined for the first destination;

generating at least one packet comprising at least a portion of the second quantity of data destined for the second destination;

transmitting a first packet destined for the first destination to the network switch;  
and

transmitting the at least one packet destined for the second destination to the network switch, before transmitting a second packet destined for the first destination.

36. A method according to claim 35, wherein the first quantity of data comprises at least a portion of a video media file.

37. A method according to claim 35, further comprising generating a plurality of said packets comprising at least a portion of the second quantity of data destined for the second destination.

38. A method according to claim 35, further comprising:  
transmitting the second packet destined for the first destination.

39. A method according to claim 35, wherein said write call further comprises a plurality of destinations.

40. A method according to claim 35, further comprising:  
communicating at least one of said packets to a network interface card.

41. A method according to claim 35, further comprising:  
performing a zero-copy write of said quantity of data.

42. A method according to claim 35, wherein said write call comprises a write vector.

43. A computer program product for balancing load on a network switch; the computer program product comprising:

a computer-readable medium comprising a program module, the program module including instructions for:

receiving a write call from an application through a socket, the write call comprising a plurality of destinations, including a first destination and a second

destination, and a first quantity of data destined for the first destination, and a second quantity of data destined for the second destination;

packetizing said first quantity of data into a plurality of packets less than or equal to a packet size, each packet destined for the first destination;

generating at least one packet comprising at least a portion of the second quantity of data destined for the second destination;

transmitting a first packet destined for the first destination to the network switch; and

transmitting the at least one packet destined for the second destination to the network switch, before transmitting a second packet destined for the first destination.

44. A system for sending data across a network, the system comprising:

a computer running an application configured to send a write call to a driver through a socket, the write call comprising a destination and a quantity of data greater than a packet size, the application further configured to perform a zero-copy write of said quantity of data to said driver; the driver configured to packetize said data into a plurality of packets less than or equal to said packet size; and

a downstream device adapted to receive at least one of said packets.

45. A system according to claim 44, further comprising a network switch coupled to the computer and the downstream device, adapted to receive at least one of said packets and route the received packet to the downstream device.

46. A system according to claim 44, wherein the downstream device comprises a down stream device has a timing requirement for receipt of data.

47. A system according to claim 45, further comprising a plurality of said computers in communication with the network switch.

48. A system according to claim 44, further comprising a plurality of downstream devices in communication with the network switch.